

Miroir Intelligent

BEAUCHAMP Valentin
VILLEMONT Guillaume

ENCADRANTS :
BOE Alexandre
VANTROYS Thomas
REDON Xavier

Nous souhaitons remercier nos encadrants de nous avoir
permis de mener à bien ce projet

Les miroirs sont passifs et ennuyants. On les regarde tous les jours et on y voit toujours la même chose : une version de soi quasiment identique mais aussi toujours plus vieille. Ce temps passé à se regarder est du temps perdu, presque gâché mais nous sommes pourtant bien obligé de passer par la salle de bain et le miroir. Et si vous pouviez, juste à la sortie de votre douche, savoir quoi porter aujourd'hui grâce à une prévision météo ? Et si vous pouviez, tout en vous coiffant, être au courant de ce qui se passe dans le monde ? Et si vous pouviez, en vous brossant les dents, planifier votre journée de rendez-vous ? C'est ce que nous vous proposons avec notre prototype de miroir intelligent !

Table des matières

Cahier des Charges.....	4
L'aspect interface web.....	5
Page Principale.....	5
Page de paramétrage.....	6
Les Widgets et JavaScripts.....	7
L'aspect Matériel.....	7
Le serveur.....	7
La façade utilisateur.....	8
Les problèmes rencontrés et solutions.....	9
Interaction utilisateur.....	9
Calibration de la matrice de LED.....	9
Conclusion.....	10
Annexe.....	11

Le sujet de notre projet de Miroir Intelligent nous a immédiatement plu pour son aspect high tech et pourtant bien ancré dans le présent. En effet aujourd'hui avec la tendance du tout numérique, le nombre d'objet connecté ne cesse de croître, cela est une bonne expérience et une satisfaction de pouvoir créer et designer son propre objet connecté. Notre projet s'est articulé en deux parties. Premièrement nous avons réalisé l'interface web qui sert d'interface utilisateur. Deuxièmement nous avons construit le cadre et configuré la partie matériel servant à héberger notre page web.

Cahier des Charges

Nous décidons de créer notre miroir de manière à ce qu'il puisse :

- Disposer d'une interface utilisateur
- Reconnaître l'utilisateur
- Être léger
- Bénéficier d'une connexion internet
- Disposer d'une source d'alimentation

Après recherche et documentation, la solution la plus simple à mettre en place est une application Web hébergée sur un serveur apache. Pour satisfaire la portabilité du miroir nous ferons tourner notre serveur sur Raspberry Pi. L'utilisateur doit pouvoir paramétrer la page web sur laquelle le miroir se connecte, afin d'afficher des informations personnalisées à chaque utilisateur. Pour satisfaire au critère d'interaction avec l'utilisateur nous utilisons une matrice de LED permettant de rendre la surface du miroir tactile.

L'aspect interface web

Page Principale

La page principale de l'application web affiche plusieurs widgets permettant à l'utilisateur d'obtenir les différentes informations désirées

On s'arrange avec le CSS pour avoir un fond noir et les écritures en blanc très tranché. Cette configuration nous permettra, une fois l'affichage dernière un miroir sans tain, de ne distinguer que les écritures et conserver l'effet réfléchissant du miroir sur les partie noire.

Cette page contient aussi un menu déroulant permettant à l'utilisateur d'activer ou désactiver les applications qu'il souhaite. Afin de rendre l'application la plus modulable possible, ce menu contient plusieurs autres menus déroulant permettant de désactiver ou d'activer seulement certaine partie de chaque widget. L'utilisateur peut ainsi choisir par exemple d'activer l'horloge sans voir la date et le calendrier. Le menu contient aussi un bouton permettant à l'utilisateur d'aller sur une page de paramétrage afin de pouvoir changer les différentes informations personnel de celui-ci (agenda, langue, localisation).



Page principale de l'interface

Page de paramétrage

Le paramétrage des widgets s'effectue initialement par l'édition d'un fichier JavaScript regroupant tous les paramètres possiblement modifiables. Nous avons développé une seconde page web PHP qui permet de modifier ces paramètres sans devoir modifier le fichier texte JavaScript de configuration. Pour cela nous avons été amenés à modifier ce fichier JS pour qu'il aille chercher ses valeurs JavaScript dans des fichiers textes correspondant. Ce choix, qui n'est pas le plus 'propre' ni même le plus simple, a été motivé par le désir de portabilité de l'application. Nous aurions pu déployer une base SQL regroupant les paramètres et les utilisateurs, cependant cela alourdissait l'installation de l'application ainsi que sa complexité.

Cette page de paramétrage permet donc à l'utilisateur d'entrer ses préférences dans l'application du miroir. Elle est constituée de plusieurs champs, chaque champ correspondant à un paramètre, qui peut être rempli par l'utilisateur. Nous avons pensé à plusieurs manières d'interagir avec l'utilisateur mais nous avons bénéficié d'une interface tactile nous donnant la possibilité d'intégrer un autre widget : une application clavier numérique. Ainsi, à la sélection du champ à modifier, un petit clavier apparaît et l'utilisateur peut simplement taper ce qu'il désire.



Page de paramétrage de l'interface

Les Widgets et JavaScripts

L'application est constituée de plusieurs widgets affichants différentes informations. Comme décrit précédemment, chaque widget est subdivisé en partie désactivable indépendamment.

- Horloge : Ce widget récupère la date et l'heure de la raspberry pi et les affiche sur le miroir. La date et l'heure sont désactivables indépendamment.
- Le calendrier : Il récupère différentes informations sur l'agenda google de l'utilisateur, qui peut entrer l'adresse de celui-ci afin de connaître ses événements à venir. Il y a aussi la possibilité de charger des calendriers préenregistrés, comme le calendrier lunaire. L'utilisateur peut régler le nombre d'évènement que le miroir va afficher.
- Météo : La météo se connecte au site OpenWeatherMap.org afin de récupérer les informations de température, temps, vitesse du vent, lever et coucher du soleil. Le Widget affiche aussi des prévisions météo sur un nombre de jour que l'utilisateur peut régler. On offre la possibilité de désactiver séparément la température actuelle, la prévision météo des jours à venir, et les informations complémentaires (vitesse du vent...).
- News : Les nouvelles du jour s'affiche sur le miroir au travers de cette application. Le widget récupère le flux RSS de votre journal préféré, par défaut Le Monde, et affiche les titres des articles.
- Compliment : Ce widget permet à l'utilisateur d'entrer des phrases que le miroir affichera selon un créneau horaire choisis, matin, après-midi et soir.
-

L'aspect Matériel

Le serveur

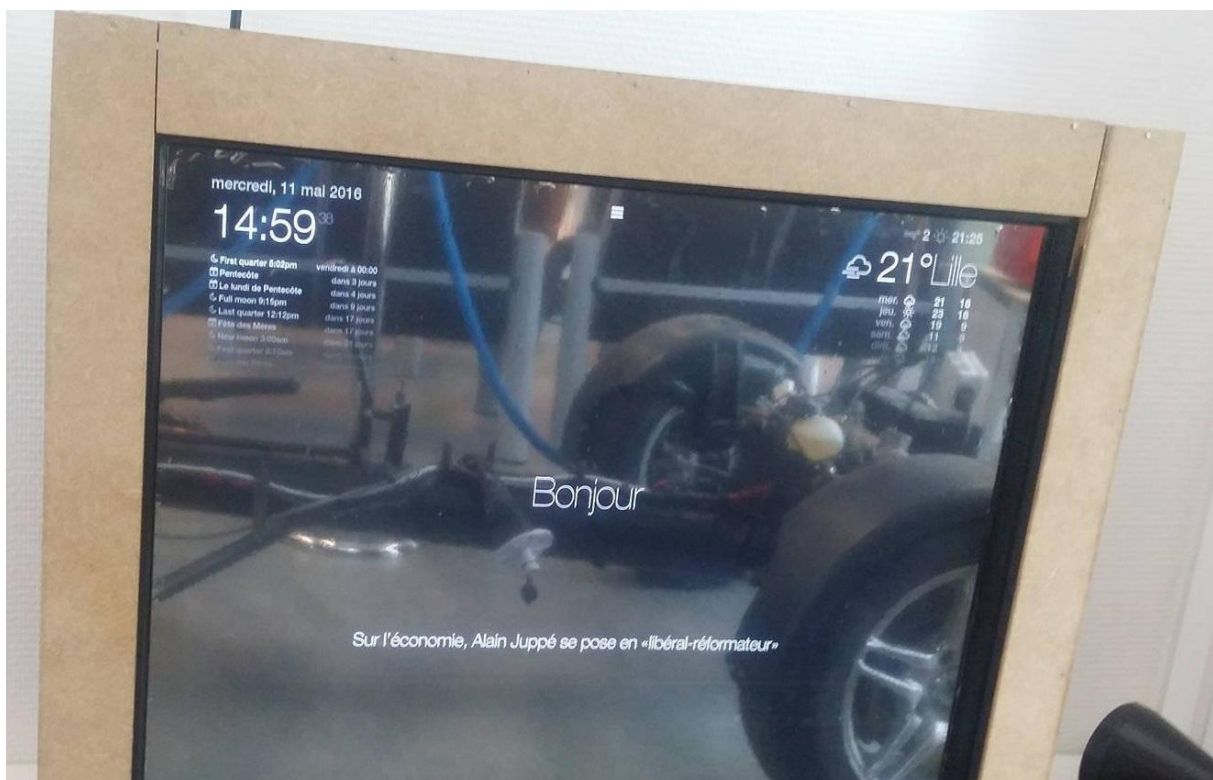
Pour héberger notre page web nous utilisons une Raspberry Pi. Pour installer notre interface, il suffit d'installer un serveur apache. Dans les paramètres de l'OS on indique de lancer le navigateur web au démarrage. On met 'localhost' en page de démarrage du navigateur de manière à afficher directement la page d'accueil de l'interface avec les widgets. On indique aussi dans les paramètres du navigateur de se lancer en mode kiosk, ce qui le passe en plein écran, pour des questions esthétiques.

La façade utilisateur

Produit fini, l'utilisateur se retrouve devant un miroir enfermé dans un cadre, lui affichant des informations. Le cadre est crucial et sert à maintenir le miroir, la matrice de LED tactile, et l'écran relié à la raspberry ensemble.

Pour réaliser le cadre du miroir, nous avons demandé de l'aide au Fabricarum, qui nous a laissé utiliser la découpeuse laser afin de découper les planches nécessaire à sa création.

Nous avons aussi bénéficié de la matrice de LED d'un projet précédent. Cette matrice est en fait un cadre sur lequel des LED et des capteurs infrarouges sont disposés. Ainsi toute la surface intérieure du cadre est quadrillée par des faisceaux infrarouges. Cette structure supporte le miroir et les faisceaux sont à ras du miroir. Ainsi lorsque l'on touche le miroir pour cliquer, les faisceaux sont coupés, et les coordonnées X et Y du point de pression est alors connu. Nous disposons ainsi d'une interface tactile. Cette matrice est la version Q3 de la marque PQLab. Nous utilisons les drivers constructeurs pour calibrer l'interface tactile.



Miroir fini

Les problèmes rencontrés et solutions

Interaction utilisateur

L'interaction utilisateur a été notre principale préoccupation. Il fallait pouvoir offrir un moyen simple et intuitif de régler les paramètres. Nous désirions aussi donner plus de possibilité à notre miroir, tel que de l'image augmentée en ajoutant des éléments en temps réel sur la réflexion sur miroir. Nous avons songé dans un premier temps à lancer un serveur VNC sur la raspberry Pi de manière à avoir accès directement au navigateur web et ses paramètres. Puis nous nous sommes lancés dans la reconnaissance faciale. Cela nous aurait permis à la fois de reconnaître l'utilisateur et par un système de gestuel de configurer le miroir, et à la fois de fournir une application d'image augmentée en temps réel. Cependant la configuration n'aurait pas pu être aussi précise qu'avec une modification au clavier. Le développement de cette idée trop chronophage a été arrêté. L'interface tactile nous a fourni à merveille la solution de configuration mais l'idée de modification en temps réel n'a pas abouti.

Une amélioration possible avec un capteur de présence et/ou une balise bluetooth pourrait permettre de sortir le miroir d'un mode veille et de reconnaître l'utilisateur.

Calibration de la matrice de LED

Nous avons passé beaucoup de temps avec la matrice tactile. Nous souhaitons héberger tout notre projet sur raspberry pi, cependant nous nous sommes rendu compte que le driver de calibration requérait l'usage d'Adobe AIR et bien que la matrice soit reconnu comme une souris classique par n'importe quel appareil, Adobe ne fournit de version d'AIR pour les architecture arm comme celle de la raspberry. La calibration sur la raspberry était donc impossible. Nous utilisons donc un ordinateur pour pouvoir utiliser la matrice tactile. Bien qu'il semblerait que la calibration puisse se faire sur l'ordinateur puis l'utilisation sur raspberry pi, nous n'avons pas pu expérimenter avec succès cette configuration.

Conclusion

Dans un société où les objets connectés sont de plus en plus présents, la réalisation de notre miroir intelligent nous a permis de découvrir le travail de réflexion qu'il doit y avoir derrière chaque objet.

Bien que nous n'ayons pas complété tous les objectifs que nous nous étions fixés, tel que la reconnaissance de l'utilisateur, nous sommes heureux d'avoir pu présenter un objet fini esthétiquement et fonctionnel.

Ce projet a été l'occasion de découvrir de nouveau langage par nous même, notamment le JavaScript et le PHP. Nous avons vu ce qu'est la réalisation d'un projet jusqu'à sa finition sur une période de temps assez longue.

Annexe

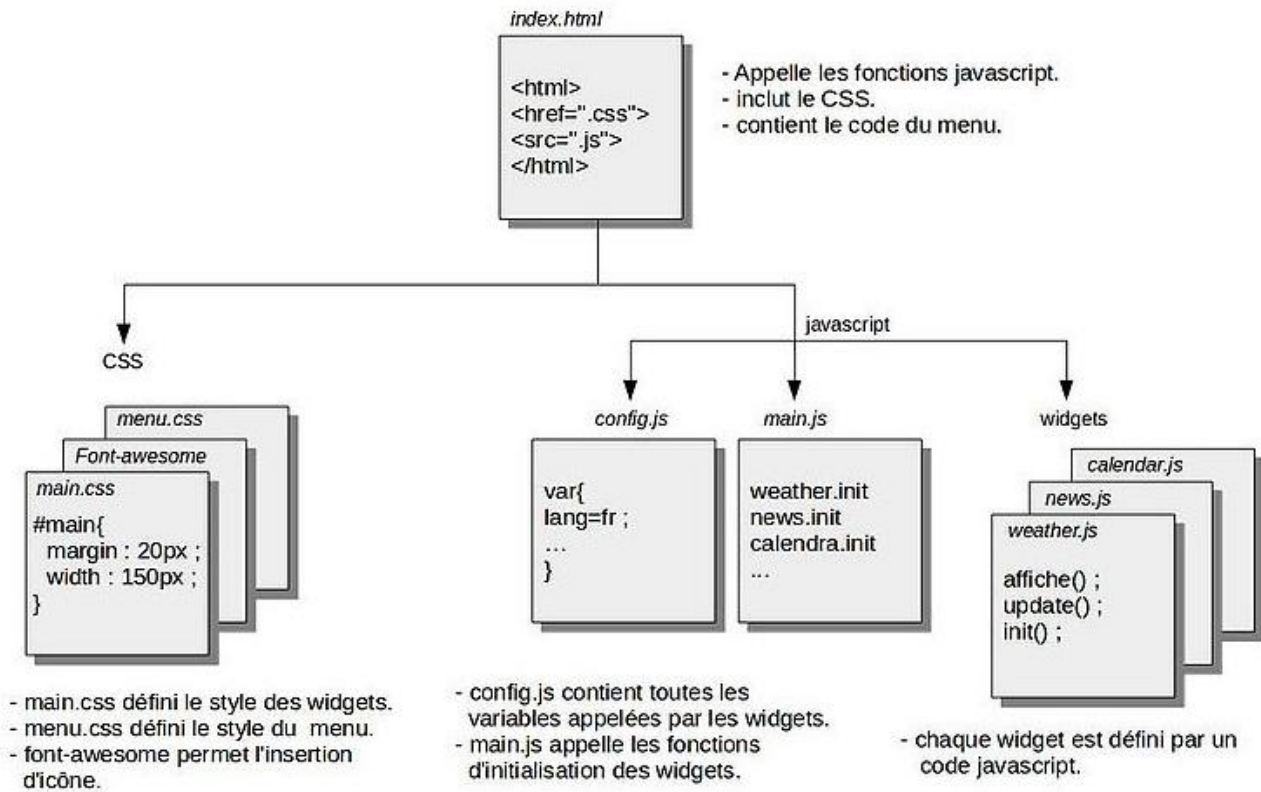


Schéma d'interaction des différents .js et php

Vous pouvez trouver le code de ce projet à l'adresse :

<https://github.com/vbeaucha/miroir-intelligent>